TITLE OF THE INVENTION

System and Method of Automatic Object Classification by Tournament Strategy.

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Application Ser. No. 60/283,635, filed April 16, 2001, entitled "Device and Method for General Classification of Objects Based on Selection Procedure Applied to Object Pairs," and incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

The present invention relates to automatic classification of objects in general, and more particularly to automatic defect classification.

BACKGROUND OF THE INVENTION

Automatic object classification is an increasingly important aspect of many industrial systems. For instance, automatic defect classification (ADC) is important aspect of semiconductor production. In conventional classification systems, classification rules are derived from a learning set of reference objects (e.g. defect images in ADC) and then applied in a production environment.

One of the most difficult and important stages in object classification is choosing an optimal set of formal features to form a feature space. The feature space should not only describe the objects of classification, but it should also relate to those object properties which best discriminate objects between different classes. Moreover, feature space selection may impact the balance between precision and generality, known to be a difficult aspect of any pattern recognition problem, as it has been shown that greater generalization may be achieved with simple rules containing a relatively small number of features, but at the expense of precision when defining rules for the learning set.

SUMMARY OF THE INVENTION

The present invention provides a system and method of automatic object classification that overcomes disadvantages of the prior art. A novel technique for the automatic generation and application of object classification rules is described. A binary rule is defined for every pair of different defined classes C_i,C_j, where i,j = 1,2,...,n; i<j; and where n is the number of defined classes, resulting in n*(n-1)/2 binary rules for n classes. A binary rule for classes C_i and C_j is generated in such a way that it discriminates between these classes only. Thus, when relating a binary rule to pair of classes C_i, C_j, an object O is classified either as belonging to class C_i or as belonging to class C_j. A tournament strategy of classification is then employed where for every pair of classes a winning class is found to which the object most likely belongs. The class which wins the most times then becomes the ultimate winner and thus the class among all other classes to which the object most likely belongs.

For specific types of binary rules, such as where a fuzzy-logic calculation mechanism is used, a binary rule for class pair (C_i, C_j) may be represented in the form of two fuzzy classification rules R_{ij} and R_{ji}. Thus, for every object O, values $R_{ij}(O)$ and $R_{ji}(O)$ may characterize a fuzzy degree of belonging of object O to classes C_i and C_j respectively. Thus, when relating a pair of rules R_{ij}, R_{ji} to pair of classes C_i, C_j, object O will be classified either as belonging to class C_i (if $R_{ij}(O) > R_{ji}(O)$), or as belonging to class C_j (if $R_{ji}(O) > R_{ij}(O)$).

In one aspect of the present invention a system for automatic object classification is provided including means for applying a plurality of binary rules to an object, where any of the binary rules is operative to classify the object to one of a pair of classes, and means for determining to which of the classes the object is classified the greatest number of times subsequent to the application of the binary rules.

In another aspect of the present invention the system further includes means for automatically generating the binary rules.

In another aspect of the present invention the system further includes a learning set having a plurality of the objects, where each of the objects in the learning set is pre-

classified as belonging to one of the classes, and where the means for automatically generating is operative to generate the binary rules using the learning set.

In another aspect of the present invention the means for automatically generating is operative to generate using supervised learning.

In another aspect of the present invention each of the binary rules includes a first part and a second part, the means for determining is operative to calculate using the first part a degree of belonging of the object to one of the classes in the class pair, the means for determining is operative to calculate using the second part a degree of belonging of the object to the other of the classes in the class pair, and the means for applying is operative to select one of the classes in the class pairs to which the degree of belonging of the object is greater.

In another aspect of the present invention each of the parts includes at least one fuzzy logic formula including at least one named predicate related to a numerical characteristic of one of the objects, and where the means for determining is operative to calculate the degrees of belonging using the fuzzy-logic formulae.

In another aspect of the present invention the objects are images.

In another aspect of the present invention the objects are semiconductor defect images and where the classes describe defect classes for application in semiconductor production.

In another aspect of the present invention a method is provided for automatic object classification including applying a plurality of binary rules to an object, where any of the binary rules is operative to classify the object to one of a pair of a plurality of classes, and determining to which of the classes the object is classified the greatest number of times subsequent to the application of the binary rules.

In another aspect of the present invention the method further includes pre-classifying a plurality of objects in a learning set as belonging to one of the classes, and automatically generating the binary rules using the learning set, where any of the binary rules of any of the pairs of classes is generated using any of the objects in the learning set that are pre-classified as belonging to the pair of classes.

In another aspect of the present invention the automatically generating step includes generating using supervised learning.

In another aspect of the present invention the determining step includes calculating a degree of belonging of the object to one of the classes in the class pair using a first part of each of the binary rules, the determining step includes calculating a degree of belonging of the object to the other of the classes in the class pair using a second part of each of the binary rules, and the applying step includes selecting one of the classes in the class pairs to which the degree of belonging of the object is greater.

In another aspect of the present invention the determining step includes calculating the degrees of belonging using a fuzzy-logic formula included in each of the parts and including at least one named predicate related to a numerical characteristic of one of the objects.

The disclosures of all patents, patent applications, and other publications mentioned in this specification and of the patents, patent applications, and other publications cited therein are hereby incorporated by reference in their entirety.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated more fully from the following detailed description taken in conjunction with the appended drawings in which:

Fig. 1A is a simplified block flow illustration of a supervised method of generating classification rules in an object classification system, operative in accordance with a preferred embodiment of the present invention;

Fig. 1B is a simplified block diagram of interaction between a user and system components for generating classification rules in an object classification system, operative in accordance with a preferred embodiment of the present invention;

Fig. 1C is a simplified block diagram of interaction between a user and system components for applying generated classification rules, operative in accordance with a preferred embodiment of the present invention;

Fig. 2 is a simplified flowchart illustration of a method of generating tournament classification rules in an object classification system, operative in accordance with a preferred embodiment of the present invention; and

Fig. 3 is a simplified flowchart illustration of a method of applying tournament classification rules in an object classification system, operative in accordance with a preferred embodiment of the present invention.

GLOSSARY OF TERMS

The following terms are used throughout the specification and claims and are defined as follows:

AOCS (Automatic Object Classification System): a system for describing classes of objects (e.g., microchip layer defect images) and automatically classifying similar objects.

Object: A named unique entity (e.g., a microchip defect image) that can be analyzed according to specific features.

Class: A set of objects that are related to a unique class name, provided by the user.

Classification: The process and result of manually and/or automatically providing class names to groups of objects.

Learning set: A set of objects, typically manually classified by a user and applied for building rules for automatic classification of objects.

Feature: A named real function of an object.

Predicate: A named real function of an object, which has values belonging to interval [0,1].

Crisp predicate: An expression in the form f > n or f < n, where f is name of a feature (for which normalization is not defined), and n is a real number. Like an ordinary predicate, a crisp predicate defines numerical function of objects as follows:

If crisp predicate p is of the form (f > n) and f(O) > n then p(O) = 1;

If crisp predicate p is of the form (f > n) and f(O) < n then p(O) = 0;

If crisp predicate p is of the form (f < n) and f(O) > n then p(O) = 0;

If crisp predicate p is of the form (f < n) and f(O) < n then p(O) = 1;

where O is an object.

Examples of Crisp predicates include:

Dimension > 50

where Dimension is a feature name.

Transformation of features into predicates: The process of forming a predicate from a feature. This can be done in either of the following ways:

1) By applying a special function;

2) By applying a statistical normalization.

Modifier: An expression such as Somewhat, Not, Not Somewhat, More-or-Less, Not More-or-Less.

Predicate with modifier: An expression in the form <Modifier> <Predicate>, defined for non-crisp predicates only. For every non-crisp predicate P, modifiers change the function of the predicate. For example:

Not P(O) = 1 – P(O);

Somewhat P(O) = sqrt(P(O));

where O is an object. Other functions may be applied as operators for implementation of modifiers.

Or-predicate: An expression in the form $P_1|P_2|\ldots|P_n$, where $P_1, P_2, \ldots, P_n$ are predicates which may contain modifiers or be crisp predicates, and n is a natural number. An or-

predicate defines a real function of objects. For example: Let predicate P be an Or-predicate ($P_1|P_2| \ldots P_n$). For every object O, $P(O) = \max((P_1(O),P_2(O),..., P_n(O))$.

And-predicate: An expression in the form $P_1\&P_2\& \ldots \&P_n$, where $P_1,P_2,\ldots, P_n$ are predicates, which may contain modifiers or be crisp predicates, or Or-predicates, and n is a natural number. An and-predicate defines a real function of objects. For example: Let predicate P be an And-predicate ($P_1\&P_2\& \ldots \&P_n$). For every object O, $P(O) = \min((P_1(O),P_2(O),..., P_n(O))$.

Rule: A predicate, which may contain modifiers or be crisp predicate, or-predicate, or and-predicate. For a rule R and an object O, a rule value is designated as R(O). Rule value R(O) characterizes the degree of belonging of an object O to an object class as defined by rule R.

Examples of rules:

1) A simple rule containing one predicate:

R_1 = Circular

2) A rule formed from an and-predicates:

R_2 = Black & Not Circular & (Dimension > 50).

Here Black and Circular are predicate names, Dimension is a feature name, and Dimension > 50 is a crisp predicate.

3) A rule which contains an or-predicate:

R_3 = Circular & (Black | Dimension > 50).

Here Circular and Black are predicate names, and (Black | Dimension > 50) is an or-predicate.

Degree of belonging: A numerical value (e.g., between 0 and 1), which characterizes a fuzzy value of classification of an object O in a class C.

Types of classification results: A characterization of a degree of belonging, for example: Belonging to class C_i, Unknown, Cannot decide.

Belonging to class C_i: A classification result for object O if:

a) Its maximal degree of belonging for object O relates it to C_i.

b) Its degree of belonging is greater than a certain threshold.

c) The difference between its degree of belonging and the maximal degree for other classes is greater than a certain threshold.

Unknown: A classification result for object O if its maximal degree of belonging to defined classes is less than certain threshold.

Cannot decide: A classification result for object O if:

a) Its maximal degree of belonging is greater than certain threshold;

b) The difference between its degree of belonging and the maximal degree for other classes is less than certain threshold.

Error function h(R,C1,C2): The numerical characteristics of errors which arise when applying rule R for discrimination between class C1 and class C2 in a given learning set. An error function may be calculated as follows:

h(R,C1,C2) = n11 / (n11 + n12) + n22 / (n22 + n21),

where

n11 = number of objects O, classified as class C1 by the user and as class C1 by AOCS;

n12 = number of objects O, classified as class C1 by the user and as class C2 by AOCS;

n21 = number of objects O, classified as class C2 by the user and as class C1 by AOCS;

n22 = number of objects O, classified as class C2 by the user and as class C2 by AOCS.

Note: Class C2 also may represent all objects not belonging to class C1.

Winning strategy of classification. Given two or more rules R_1, R_2, ..., describing classes C_1, C_2, ... correspondingly, an object O is classified as belonging to the class for

which its degree of belonging (i.e. related rule value) is maximal. The corresponding class and rule are called the winning class and rule.

Binary rule: A classification rule related to a pair of different classes (C_i,C_j) only. Given an object O, a binary rule classifies it either as belonging to C_i or C_j. For example, a binary rule for classes (C_i,C_j) consists of a rule pair (R_{ij},R_{ji}), where rule R_{ij} discriminates objects of class C_i from class C_j and rule R_{ji} discriminates objects of class C_j from class C_i. Rules R_{ij} and R_{ji} may include predicates as described above, and a comparison of values $R_{ij}(O)$ and $R_{ji}(O)$ determines the class to which object O belongs according to the winning strategy of classification.

Tournament strategy of classification. Given two or more classes C_1, C_2, ... C_n, an object O is classified in tournament fashion where, for every pair of classes, a winning class is determined as being the class which wins the most times using binary rules.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Reference is now made to Fig. 1A, which is a simplified block flow illustration of a method of generating classification rules in an automatic object classification system, operative in accordance with a preferred embodiment of the present invention. In the method of Fig. 1A supervised learning is used in which, after every learning step (i.e. after changing a classification rule) the rules are applied to objects of a learning set, and their classification results are compared with classifications of the objects that are *a priori* known to be correct. After this comparison, specific rule changes may be effected, in case of successful learning, or prevented in consideration of alternative rule changes. Evaluation of success or failure may be performed with respect to the number of classification errors for all objects in the learning set.

Reference is now made to Fig. 1B, which is a simplified block diagram of a method of interaction between a user and system components for generating classification rules in an object classification system, operative in accordance with a preferred embodiment of the present invention. In Fig. 1B a user provides, typically manually, *a*

priori correct class names for reference objects and thus creates a learning set. The learning set is then used for creation of classification rules according to method of Fig. 1A and/or Fig. 2, described in greater detail hereinbelow.

Reference is now made to Fig. 1C, which is a simplified block diagram of a method of interaction between a user and system components for application of generated classification rules operative in accordance with a preferred embodiment of the present invention. In Fig. 1C a user obtains the classification of an object. Classification is carried out using classification rules obtained according to the method of Fig. 1B. The automatic object classifier operates according to method of Fig. 3, described in greater detail hereinbelow.

Reference is now made to Fig. 2, which is a simplified flowchart illustration of a method of supervised generation of tournament classification rules in an automatic object classification system, operative in accordance with a preferred embodiment of the present invention. In a tournament classification system a binary rule is generated for every pair of different defined classes, where a binary rule for classes C_i, C_j may include a pair of rules, R_{ij} and R_{ji}. When relating a pair of rules R_{ij}, R_{ji} to classes C_i, C_j, an object O will be classified as belonging to class C_i if $R_{ij}(O) > R_{ji}(O)$, and as belonging to class C_j if $R_{ij}(O) < R_{ji}(O)$.

In the method of Fig. 2 n defect classes are defined with each class C_i (where $1 \leq i \leq n$) having $n - 1$ rules $R_{i1}, R_{i2}, \ldots, R_{i,i-1}, R_{i,i+1}, \ldots, R_{in}$. The role of every such rule R_{ij} is to discriminate objects of class C_i from objects of class C_j. For all pairs (C_i, C_j) of defined classes the corresponding rule pairs (R_{ij}, R_{ji}) are created as follows:

1) Transform features into predicates.

2) Build initial rule pair table.

3) Improve rule pairs.

Step 1, Transformation of features into predicates, is described hereinabove. Steps 2 and 3 are now described in greater detail.

2) Build initial rule pair table. The rules may be organized in a table as is shown in Table A below.

	C_1	C_2	C_3	...	C_n
C_1	–	R_{21}	R_{31}	...	R_{n1}
C_2	R_{12}	–	R_{32}	...	R_{n2}
C_3	R_{13}	R_{23}	–	...	R_{n3}
...	...	...	...	...	...
C_n	R_{1n}	R_{2n}	R_{3n}	...	--

TABLE A

In Table A every box related to class C_i is populated by copies of a rule $R(C_i)$ which may be obtained by conventional fuzzy-logic methods based on the winning strategy of classification. Thus, $R_{i1} = R_{i2} = \ldots = R_{in} = R(C_i)$; i = 1,2, ..., n. Rules R_{ii} in the diagonal boxes are made empty. Alternatively, all boxes of Table A may be initially made empty. In this case, all initial rules R_{ij} are considered to be empty. Non-empty initial rules may help in avoiding local minima of the error function applied for improvement of rule pairs.

3) Improve rule pairs. For every pair of defined classes (C_i, C_j), where i > j, the corresponding pair of rules R_{ij} and R_{ji} may be improved. Arrays P_{ij} and P_{ji} of prospective predicates for inclusion into rules R_{ij} and R_{ji} are formed as follows. For every defined predicate p (with or without a modifier), average predicate values A_{pi} and A_{pj} may be calculated where A_{pi} is the average value of p for objects of class C_i and A_{pj} is the average value of p for objects of class C_j. It may be seen that predicates with small average values are not desirable as prospective predicates for being and-predicates in rules since a) the minimum value of all and-predicates forms the rule value and b) for rule improvement the greatest possible rule values are sought. Therefore, a threshold constant T_p may be defined such that only predicates with an average value grater than T_p are included into the arrays P_{ij} and P_{ji}. Typically a value of T_p=0.6 is believed to be suitable for filtering out predicates with small average values. Thus, for every predicate p, if $A_{pi} > T_p$ then p is included into array P_{ij}, and if $A_{pj} > T_p$, then p is included into array P_{ji}. A predicate may also be considered as more prospective for inclusion where the predicate has a larger difference of average values for classes C_i and C_j. A value T_{pij} which characterizes the power of every predicate p for discriminating classes C_i and C_j may be then calculated by conventional statistical methods.

Both arrays P_{ij} and P_{ji} may then be sorted by descending of value T_{pij}. A constant K may be defined such that only the first K elements of arrays P_{ij} and P_{ji} are kept, and all other elements removed. The value of K is preferably set in accordance with effectiveness and efficiency considerations. Typically, a value of K=20 is believed to provide satisfactory results and may be increased for achieving still better classification at the expense of rule generation time.

Once arrays P_{ij} and P_{ji} have been constructed, the rules R_{ij} and R_{ji} may be improved by applying an oscillation algorithm for finding optimal rule sets. Each rule pair is evaluated using the error functions $h(R_{ij},C_i,C_j)$ and $h(R_{ji},C_j,C_i)$ calculated using the winning strategy of classification for two classes only. The oscillation algorithm is then used in parallel for rules R_{ij} and R_{ji}, performing one forward and one backward step for R_{ij}, and then performing the same for R_{ji}, and so on. In a forward oscillation step predicates are added to the rule. Each next predicate P[I] which does not yet belong to rule R is taken from the associated array P. Rule R $\cup$ P[I] is tested. If $h(R \cup P[I]) > h(R)$, then predicate P[I] is added to rule R. Otherwise, proceed to element P[I+1]. This procedure is repeated up to the end of array P. In a backward oscillation step predicates are deleted from rule R. The removal of the first predicate p from rule R is attempted. If the resulting rule has a better error function, then predicate p may be removed from rule R, otherwise predicate p remains in rule R. The removal of the second predicate p from rule R is attempted, and so on, until all predicates in rule R are considered. The oscillation algorithm terminates if no change results after performing the backward oscillation step, or once a predetermined number of the oscillation steps are performed.

Once the oscillation algorithm has been applied for all rule pairs, a table may then be constructed from the optimal rule pairs R_{ij} and R_{ji} for every pair of different classes C_i and C_j.

Reference is now made to Fig. 3, which is a simplified flowchart illustration of a method of applying tournament classification rules in an object classification system, operative in accordance with a preferred embodiment of the present invention. In the method of Fig. 3 the results of the method of Fig. 2 are applied to an object using a

Tournament strategy of classification. A maximum of n(n–1)/2 binary rules are applied, distinguishing between classes C_i and C_j, where i, j < n, and where n is number of classes defined in the learning set. Preferably, these rules are applied to objects whose expected classes are the same as those in the learning set.

In the method of Fig. 3 the following steps are performed:

1) Prepare a table of initial tournament values.

2) Apply binary rules and correct tournament values.

3) Sort the tournament values.

4) Determine the winning class for the object.

Each of these steps are now described in greater detail.

1) Prepare a table of tournament values. A table is preferably prepared as is shown in Table B below.

C_1	C_2	C_3	...	C_n
V_1	V_2	V_3	...	V_n

TABLE B

In Table B n is the number of defined classes. Initially, all the values V_1,V_2,V_3, ..., V_n are typically set to 0.

2) Apply binary rules and correct tournament values. For the object O being classified, binary rules related to class pairs (C_i, C_j) for all i=1,2,...,n, j=1,2,...,n, i<j are applied for all class pairs, preferably sequentially. After every such application V_i is increased by a fixed amount, typically 1, if class C_i wins, or V_j is increased by the same fixed amount if class C_j wins. Where the calculation results in a classification type of "cannot decide" or "unknown" V_i and V_j are left unchanged. If the binary rules are in the form of rule pairs (R_{ij},R_{ji}) then rule values $R_{ij}(O)$ and $R_{ji}(O)$ for all i=1,2,...,n, j=1,2,...,n, i<j are calculated for all rule pairs. After every step in this calculation V_i is increased by a fixed amount, typically 1, if $R_{ij}(O) > R_{ji}(O)$, or V_j is increased by the same fixed amount if $R_{ji}(O) > R_{ij}(O)$.

3) Sort the tournament values. The resulting tournament values [V_1, V_2,..., V_n] are sorted in descending order. An analysis of the three maximal values V_i, V_j, V_k is then performed as follows.

4) Determine the winning class for the classified object. For classes C_i, C_J, and C_k:

a) If $V_i > V_J = V_k$ (i.e. exactly one maximum value exists), then C_i is the winning class.

b) If $V_i = V_j > V_k$ (i.e. exactly two maximum values V_i and V_J exist), then:

C_i is the winning class if $R_{ij}(O) > R_{Ji}(O)$;

C_J is the winning class if $R_{Ji}(O) > R_{iJ}(O)$;

c) If $V_i = V_J = V_k$ (i.e. more than two maximum values exist), then no winning class is selected.

The present invention is thus advantageous over the prior art in that a simple and effective feature space is provided for every binary rule, thus ensuring better utilization of *a priori* information and better generalization of the rules in a testing environment. The present invention also provides more precise classification of the objects. The evaluation of a predicate which separates two homogeneous classes only as in the present invention, is simpler than the evaluation of a predicate which separates a given class from all other classes as in traditional automatic object classification systems. Design and application of new predicates is also simplified for the user who needs only to consider two classes at a time with the present invention, rather than approach design and application as a multi-class problem.

It is appreciated that one or more of the steps of any of the methods described herein may be omitted or carried out in a different order than that shown, without departing from the true spirit and scope of the invention.

While the methods and apparatus disclosed herein may or may not have been described with reference to specific hardware or software, it is appreciated that the methods and apparatus described herein may be readily implemented in hardware or software using conventional techniques.

While the present invention has been described with reference to one or more specific embodiments, the description is intended to be illustrative of the invention as a whole and is not to be construed as limiting the invention to the embodiments shown. It is appreciated that various modifications may occur to those skilled in the art that, while not specifically shown herein, are nevertheless within the true spirit and scope of the invention.